Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) A process for managing, in an interrupt stage, a memory stack associated with a microcontroller, having a number of interrupts, according to a Program Counter signal and to a Condition Code Register signal that includes bits, that can be contained in respective registers, the process comprising the operations of:

providing a first part of memory stack, comprising a register for said Program Counter signal, and a second part of memory stack, made up of a bank of memory elements equal in number to the bits of said Condition Code Register signal times the number of interrupts of the microcontroller; and

causing said two parts of memory stack to function in parallel by employing respective stack-pointer signals.

2. (Original) The process according to claim 1, further comprising generating said respective stack-pointer signals with values equal to first memory locations available within the respective part of memory stacks.

3. (Original) The process according to claim 1 wherein, in order to drive said two parts of stack, the process comprises the operation of generating:

a first signal, for discriminating between a push operation and a pop operation;

a second signal, for explicitly enabling the push operation or pop operation on said first part of memory stack; and

a third signal for explicitly enabling the push operation or pop operation on said second part of memory stack.

4. (Original) The process according to claim 3, wherein, when an interrupt occurs, the process comprises the operations of:

setting said first signal so as to obtain a push operation;

driving signals for writing said two parts of memory stack by setting said second signal and said third signal so as to decrement the respective stack pointers, so as to point to next available memory locations.

5. (Original) The process according to claim 3 wherein, in order to restore the state at the end of an interrupt, the process comprises the operations of:

setting said first signal so as to obtain a pop operation;

driving said second signal and said third signal so as to increment the stack pointers of said first part and said second part of stack, so as to point to a location containing a stored state to be restored; and

setting the signals for carrying out reading of said two parts of memory stack.

6. (Original) A stack management device associated with a microcontroller having a number of interrupts, the device comprising:

a first memory stack comprising a register for a Program Counter signal;

a second memory stack for a Condition Code Register signal that includes a number of bits, the second memory stack being made up of a bank of memory elements equal in number to the bits of said Condition Code Register signal times the number of interrupts of the microcontroller; and

at least one manager module configured to cause said first and second memory stacks to function in parallel by respective stack pointer signals.

7. (Original) The device according to claim 6, the at least one manager module is configured to generate said respective stack pointer signals with respective values equal to a first memory location of the first memory stack and a first memory location of the second memory stack, respectively.

8. (Original) The device according to claim 6 wherein said at least one manager module is configured for generating:

a first signal for discriminating between a push operation and a pop operation;

a second signal for explicitly enabling the push operation or pop operation on said first memory stack; and

a third signal for explicitly enabling the push operation or pop operation on said second memory stack.

9. (Original) The device according to claim 8, wherein said at least one manager module is configured for:

setting said first signal so as to obtain a push operation in response to an interrupt;

driving signals for writing said memory stacks; and

setting said second and said third signal so as to decrement the respective stack pointers so as to point to next available memory location in the memory stacks, respectively.

10. (Original) The device according to claim 8 wherein, in order to restore a state at an end of an interrupt, said at least one manager module is configured for:

setting said first signal so as to obtain a pop operation;

driving said second signal and said third signal so as to increment the stack pointers of said first and second stacks so as to point to respective locations containing the state to be restored; and

setting the signals for carrying out reading of said memory stacks.

11. (Original) The device according to claim 6, wherein the at least one manager module includes:

a control unit common to said first and second memory stacks; and

first and second manager modules associated, respectively, with said first and second memory stacks for sending to said memory stacks said respective stack-pointer signals according to respective signals for execution of operation generated by said control unit.

12. (Original) The device according to claim 11, wherein said control unit is configured for sending a first signal for discriminating between a push operation and a pop operation to said manager modules.

13. (Original) The device according to claim 6 wherein the second memory stack includes memory elements that are made up of flip-flops.

14. (Currently Amended) A ~~computer program product directly loadable into an internal memory of a digital computer, said computer product comprising software code portions for performing, when said product is run on a computer, the steps of~~computer-readable medium having contents that cause a computer to manage a memory stack in an interrupt stage according to a method comprising:

providing a first memory stack, comprising a register for a Program Counter signal, and a second memory stack, made up of a bank of memory elements equal in number to bits of a Condition Code Register signal times a total number of interrupts of the computer; and

causing said memory stacks to function in parallel by employing respective stack-pointer signals.

15. (Original) A stack management device associated with a microcontroller having a number of interrupts, the device comprising:

a first memory stack having a program counter register;

a second memory stack having a condition code register;

a control unit structured to output first and second control signals; and

first and second manager modules connecting the control unit to the first and second memory stacks, respectively, the first and second manager modules being structured to send to the first and second memory stacks, respectively, respective first and second stack pointer signals according to the first and second control signals, respectively, received from the control unit.

16. (Original) The device of claim 15 wherein the first memory stack is stored in a memory having a word length corresponding to individually addressable data words and the second memory stack is made up of a bank of memory elements smaller in number than the word length of the first memory stack.

17. (Original) The device of claim 15 wherein the first manager module is configured to generate the first stack pointer signal with a value equal to a first memory location of the first memory stack and the second manager module is configured to generate the second stack pointer signal with a value equal to a first memory location of the second memory stack.

18. (Original) The device of claim 15, wherein the control unit is configured for sending to the manager modules a signal for discriminating between a push operation and a pop operation.

19. (Original) The device of claim 15 wherein the first memory stack is part of a random access memory and the second memory stack includes memory elements that are made up of flip-flops.

20. (New) The computer-readable medium of claim 14, wherein the method further comprises generating said respective stack-pointer signals with values equal to first memory locations available within the respective part of memory stacks.